



Design et formes optimales (II) : Optimisation géométrique

Grégoire Allaire, François Jouve

► To cite this version:

Grégoire Allaire, François Jouve. Design et formes optimales (II) : Optimisation géométrique. Images des Mathématiques, 2009, <http://images.math.cnrs.fr/Design-et-formes-optimales-II.html>. hal-00597633

HAL Id: hal-00597633

<https://hal.science/hal-00597633>

Submitted on 1 Jun 2011

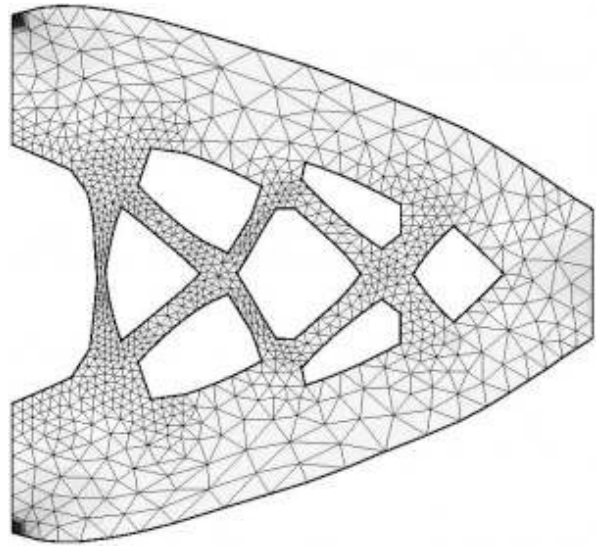
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Design et formes optimales (II)

Optimisation géométrique

Le 21 décembre 2009, par **Grégoire Allaire**
et **François Jouve**



Cet article fait suite à [celui-ci](#). Il est consacré à des avancées récentes des mathématiques et du calcul scientifique dans le domaine de l'optimisation de formes ou « optimal design ». Ces progrès ont eu des répercussions immédiates dans l'industrie (aéronautique, automobile, génie civil) en mettant à disposition des ingénieurs des logiciels d'optimisation « automatique » de design d'objets ou de structures.

Optimisation géométrique : la méthode d'Hadamard

LE problème typique de l'optimisation de forme en mécanique du solide est de trouver la forme d'une structure qui soit de rigidité maximale et de poids minimal. Mathématiquement, on pose ce problème en minimisant la compliance sous une contrainte de poids maximal. La difficulté principale vient de ce que la variable d'optimisation est la géométrie même de la structure. S'il est facile de calculer le poids d'une structure (qui est simplement proportionnel à son volume), l'évaluation de la compliance est plus délicate car elle dépend indirectement et non-explicitement de la géométrie de la structure. En effet, il faut passer par une étape (assez coûteuse) de résolution numérique par **éléments finis** des équations de Lamé pour obtenir ensuite la compliance. Mais là où les choses se compliquent sérieusement c'est lorsqu'il faut calculer la dérivée de la compliance par rapport à la forme ! Avant d'en venir à ce calcul de dérivée, expliquons pourquoi il est important en optimisation de savoir calculer des dérivées.

On apprend aux étudiants que, lorsqu'on veut déterminer les minima ou maxima d'une fonction dérivable d'une seule variable réelle, il faut les chercher parmi les points où la dérivée s'annule ainsi qu'aux bornes éventuelles de l'intervalle sur lequel on se place (on peut voir cet intervalle comme une contrainte sur la variable réelle). On procède ainsi quand on veut trouver analytiquement ces extrema. Si on veut calculer numériquement un minimum (ou un maximum), la notion de dérivée est encore très utile car elle est à la base de l'algorithme dit du **gradient** ou de la **plus grande pente** (voir bloc dépliant 3).



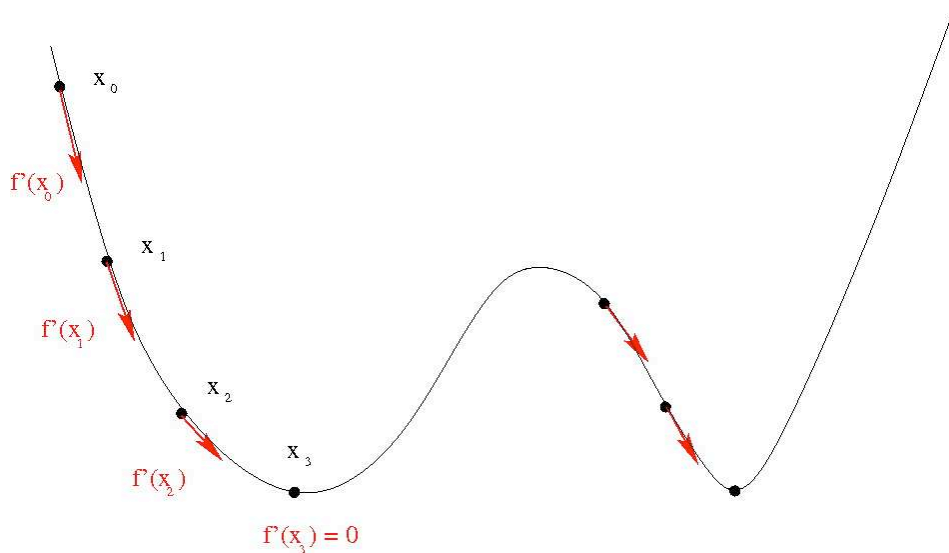
3. Algorithme du gradient ou de la plus grande pente

Il s'agit d'un algorithme itératif qui construit une suite de points $x_n, n \geq 0$ qui se rapprochent progressivement vers le minimum d'une fonction $f(x)$, définie sur \mathbb{R}^d à valeurs réelles. Cet algorithme est aussi appelé méthode de la plus grande pente car il repose sur une stratégie dont l'analogie pour un marcheur en montagne, souhaitant rejoindre le plus vite possible le fond de la vallée, est de toujours se diriger dans le sens de la plus grande pente (vers le bas !). La pente

de la fonction $f(x)$ est donnée par sa dérivée $f'(x)$, avec le signe $-$ si on veut minimiser ou descendre. A partir d'un choix initial $x_0 \in \mathbb{R}^d$ la suite de points dans \mathbb{R}^d est alors définie par la relation de récurrence

$$x_{n+1} = x_n - \delta f'(x_n) \quad \text{pour } n \geq 0,$$

où $\delta > 0$ est le pas de descente. Si ce dernier est choisi suffisamment petit, alors on peut montrer la convergence de cet algorithme vers un minimum local de la fonction f . Remarquez que des choix initiaux différents peuvent conduire à des points de minimum différents. Le point crucial dans cet algorithme est le calcul de la dérivée de la fonction que l'on veut minimiser.



Algorithme du gradient

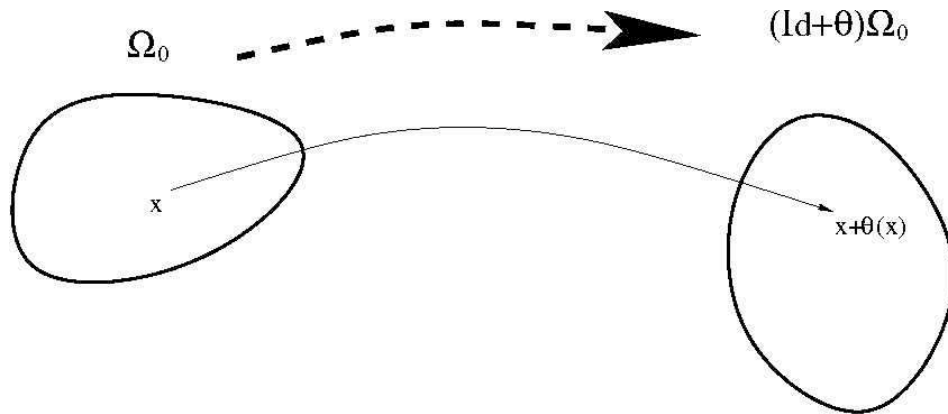
Algorithme itératif de minimisation qui peut converger vers différents minima locaux selon l'initialisation

Pour minimiser une fonction à l'aide de l'algorithme du gradient il faut donc savoir calculer sa dérivée. Mais dans notre problème de formes optimales, comment peut-on simplement définir une dérivée ? Dans son *mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées* de 1907, le mathématicien **Jacques Hadamard** a donné un cadre commode pour définir une notion de dérivation par rapport au domaine qu'on appelle aussi, depuis, la méthode d'Hadamard. Grosso modo, il s'agit de dériver par rapport à la position des points sur le bord du domaine (voir bloc dépliant 4).



4. Méthode d'Hadamard de dérivation par rapport au domaine.

L'idée de Jacques Hadamard est de fixer une forme de référence et de paramétrer des variations de cette forme initiale à l'aide d'une fonction vectorielle $\vec{\theta}$. Plus précisément, si $\Omega_0 \subset \mathbb{R}^3$ est cette forme de référence, pour toute fonction $\vec{\theta}(\vec{x})$ définie de \mathbb{R}^3 dans \mathbb{R}^3 , on définit une nouvelle forme $\Omega_\theta = (\text{Id} + \vec{\theta})\Omega_0$ comme l'ensemble des points $x \mapsto \vec{\theta}(\vec{x})$ lorsque x parcourt Ω_0 (voir la figure). Autrement dit, on paramètre la forme Ω_θ par la fonction $\vec{\theta}$. Si on se limite à de telles formes, alors la dérivée (ou le gradient) par rapport au domaine est simplement définie comme la dérivée par rapport à cette fonction $\vec{\theta}$: c'est un peu plus compliqué que de dériver par rapport à une variable réelle, mais c'est, au moins conceptuellement, une opération classique en calcul différentiel. Nous ne pouvons pas nous étendre plus sur ce sujet technique et nous renvoyons le lecteur désireux d'en savoir plus (et ayant le niveau d'une licence de mathématiques !) aux ouvrages [1] et [2].



Variation de formes "à la Hadamard"

Paramétrisation de formes par un champ de vecteur $\vec{\theta}$ à partir d'une forme de référence Ω_0

Il est important de noter tout de suite que la méthode d'Hadamard comporte une limitation essentielle : elle restreint les formes admissibles Ω_θ à avoir toutes la même **topologie** que la forme de référence Ω_0 . En effet, on peut montrer que si la fonction de paramétrisation $\vec{\theta}$ est suffisamment petite, alors Ω_θ et Ω_0 sont homéomorphes, c'est-à-dire que l'application $x \mapsto x + \vec{\theta}(x)$ est une bijection continue d'inverse continue. Concrètement en deux dimensions d'espace, cela veut dire que les formes Ω_θ et Ω_0 ont le **même nombre de trous**.



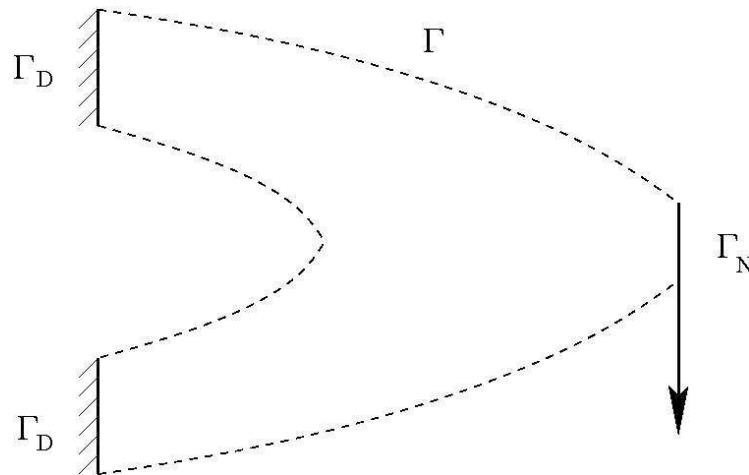
Différentes topologies en 3-d

Trois exemples de formes de topologies différentes dans l'espace à 3 dimensions : une boule (à gauche), un tore ou une bouée (au milieu), un bretzel (à droite).

En dimension trois d'espace c'est un peu plus compliqué : les formes Ω_θ et Ω_0 ont, entre autres, le même nombre de trous mais aussi d'anses ou de boucles (voir ci-dessus la différence entre une boule, un tore ou un bretzel). Cette limitation intrinsèque de la méthode d'Hadamard n'est pas que théorique mais est aussi un obstacle sérieux dans la pratique numérique.

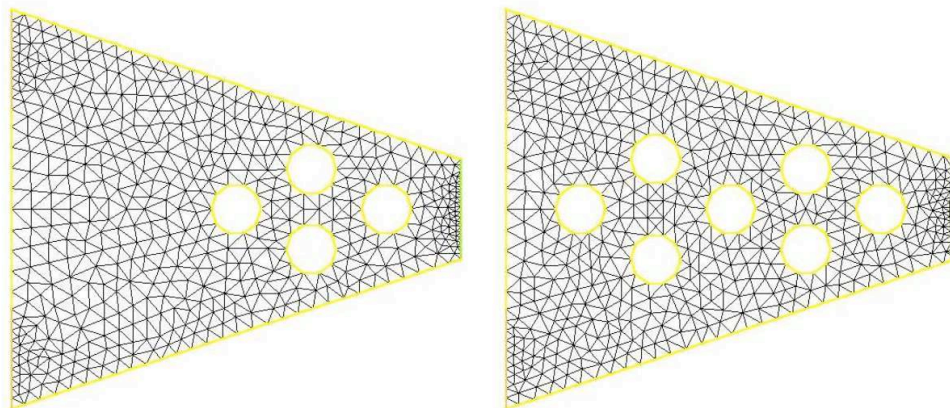
Expliquons néanmoins comment la méthode d'Hadamard est mise en œuvre dans un algorithme numérique d'optimisation géométrique de formes. Il s'agit d'un algorithme itératif qui construit, à partir d'une forme initiale, une suite de formes qui chacune améliore, par modification de la position de son bord, la performance de la précédente. Cet algorithme s'apparente à la méthode du gradient ou de la plus grande pente (voir bloc dépliant 3), c'est-à-dire que le bord de chaque forme intermédiaire est déplacé dans la direction du gradient par rapport au domaine de la fonction objectif (qui mesure la performance de la forme et que l'on veut minimiser). La convergence de l'algorithme est détectée lorsqu'on ne peut plus améliorer la performance de cette manière. Chaque itération de l'algorithme correspond, d'une part, à une résolution par éléments finis des équations de Lamé (voir **bloc dépliant 2**) dont la solution permet de calculer la performance de la forme ainsi que la dérivée par rapport au domaine, d'autre part, à une déformation du maillage, qui définit la forme, dans la direction de cette dérivée. Précisons pour le lecteur soucieux d'exactitude que, lorsque la fonction objectif n'est pas la compliance, il est nécessaire pour calculer la dérivée par rapport au domaine de cette fonction objectif d'évaluer la solution d'un autre système d'équations de Lamé, dit état adjoint (voir **bloc dépliant 1 et 2** pour plus de détails).

Nous appliquons cet algorithme à l'aide du logiciel libre **FreeFem++** dans le cas de la console optimale (le script de ce cas test est disponible dans **une boîte à outils**). Les conditions aux limites et deux maillages initiaux sont donnés dans les figures ci-dessous.



Conditions aux limites pour la console optimale

Le bord Γ_D est fixe. Une force verticale est appliquée sur le bord Γ_N . Le bord Γ est libre et est le seul soumis à optimisation.



Initialisations pour une optimisation géométrique

Deux initialisations pour le calcul d'une console optimale par la méthode d'Hadamard : avec 4 trous à gauche et 7 à droite

Les solutions successives obtenues par l'algorithme ci-dessus, pour les deux initialisations proposées, sont présentées sous forme de films ci-dessous.



Optimisation géométrique d'une console à 4 trous



Optimisation géométrique d'une console à 7 trous

Le lecteur vérifiera immédiatement que, dans chacun des films, la topologie de la forme initiale est conservée aux cours des itérations : on conserve toujours le même nombre de trous ! D'un point de vue informatique, s'il est possible de déformer un maillage (tous les logiciels d'éléments finis ne le permettent pas aisément mais **FreeFem++** si), il est très difficile, pour ne pas dire impossible, d'en changer la topologie de manière automatique (c'est-à-dire de fusionner des trous ou de casser des barres). Par conséquent, cette méthode d'optimisation géométrique fonctionne à topologie fixée et est incapable d'optimiser la topologie : la structure garde le même nombre de composantes, de bords et de trous. C'est une limitation sévère car il faut deviner la bonne topologie à imposer à la forme initiale, tâche impossible dans la plupart des cas. De plus, cet algorithme d'optimisation géométrique est très

coûteux en temps de calcul (car il peut être nécessaire de remailler au cours des itérations) et le résultat obtenu dépend fortement du choix initial de forme. C'est pourquoi les mathématiciens ont été motivés pour inventer des méthodes d'optimisation **topologique** de formes.

Pour en savoir plus, la suite **ici...**

Notes

[▲1] G. Allaire, *Conception optimale de structures*, Collection Mathématiques et Applications, Vol. 58, Springer Verlag (2007)

[▲2] A. Henrot, M. Pierre, *Variation et optimisation de formes*, Collection Mathématiques et Applications, Vol. 48, Springer Verlag (2005)

Pour citer cet article : **Grégoire Allaire** et **François Jouve**, « **Design et formes optimales (II)** » — *Images des Mathématiques*, CNRS, 2009. En ligne, URL : <http://images.math.cnrs.fr/Design-et-formes-optimales-II.html>